Network Working Group
Request for Comments: 989

John Linn (BBNCC)
IAB Privacy Task Force
February 1987

Privacy Enhancement for Internet Electronic Mail: Part I: Message Encipherment and Authentication Procedures

STATUS OF THIS MEMO

This RFC suggests a proposed protocol for the Internet community and requests discussion and suggestions for improvements. Distribution of this memo is unlimited.

ACKNOWLEDGMENT

This RFC is the outgrowth of a series of IAB Privacy Task Force meetings and of internal working papers distributed for those meetings. I would like to thank the following Privacy Task Force members and meeting guests for their comments and contributions at the meetings which led to the preparation of this RFC: David Balenson, Matt Bishop, Danny Cohen, Tom Daniel, Charles Fox, Morrie Gasser, Steve Kent (chairman), John Laws, Steve Lipner, Dan Nessett, Mike Padlipsky, Rob Shirey, Miles Smid, Steve Walker, and Steve Wilbur.

1 Executive Summary

This RFC defines message encipherment and authentication procedures, as the initial phase of an effort to provide privacy enhancement services for electronic mail transfer in the Internet. Detailed key management mechanisms to support these procedures will be defined in a subsequent RFC. As a goal of this initial phase, it is intended that the procedures defined here be compatible with a wide range of key management approaches, including both conventional (symmetric) and public-key (asymmetric) approaches for encryption of data encrypting keys. Use of conventional cryptography for message text encryption and/or authentication is anticipated.

Privacy enhancement services (confidentiality, authentication, and message integrity assurance) are offered through the use of end-to-end cryptography between originator and recipient User Agent processes, with no special processing requirements imposed on the Message Transfer System at endpoints or at intermediate relay sites. This approach allows privacy enhancement facilities to be incorporated on a site-by-site or user-by-user basis without impact on other Internet entities. Interoperability among heterogeneous components and mail transport facilities is supported.

2 Terminology

For descriptive purposes, this RFC uses some terms defined in the OSI X.400 Message Handling System Model. This section replicates a portion of X.400's Section 2.2.1, "Description of the MHS Model: Overview" in order to make the terminology clear to readers who may not be familiar with the OSI MHS Model.

In the [MHS] model, a user is a person or a computer application. A user is referred to as either an originator (when sending a message) or a recipient (when receiving one). MH Service elements define the set of message types and the capabilities that enable an originator to transfer messages of those types to one or more recipients.

An originator prepares messages with the assistance of his User Agent. A User Agent (UA) is an application process that interacts with the Message Transfer System (MTS) to submit messages. The MTS delivers to one or more recipient UAs the messages submitted to it. Functions performed solely by the UA and not standardized as part of the MH Service elements are called local UA functions.

The MTS is composed of a number of Message Transfer Agents (MTAs). Operating together, the MTAs relay messages and deliver them to the intended recipient UAs, which then make the messages available to the intended recipients.

The collection of UAs and MTAs is called the Message Handling System (MHS). The MHS and all of its users are collectively referred to as the Message Handling Environment.

3 Services, Constraints, and Implications

This RFC's goal is to define mechanisms to enhance privacy for electronic mail transferred in the Internet. The facilities discussed in this RFC provide privacy enhancement services on an end-to-end basis between sender and recipient UAs. No privacy enhancements are offered for message fields which are added or transformed by intermediate relay points. Two distinct privacy enhancement service options are supported:

1. an option providing sender authentication and integrity verification

2. an option providing sender authentication and integrity verification in addition to confidentiality service through encryption

No facility for confidentiality service in the absence of authentication is provided. Encryption and authentication facilities may be applied selectively to portions of a message's contents; this allows less sensitive portions of messages (e.g., descriptive fields)

to be processed by a recipient's delegate in the absence of the recipient's personal cryptographic keys.

In keeping with the Internet's heterogeneous constituencies and usage modes, the measures defined here are applicable to a broad range of Internet hosts and usage paradigms. In particular, it is worth noting the following attributes:

1. The mechanisms defined in this RFC are not restricted to a particular host or operating system, but rather allow interoperability among a broad range of systems. All privacy enhancements are implemented at the application layer, and are not dependent on any privacy features at lower protocol layers.

2. The defined mechanisms offer compatibility with non-enhanced Internet components. Privacy enhancements will be implemented in an end-to-end fashion which does not impact mail processing by intermediate relay hosts which do not incorporate privacy enhancement facilities. It is necessary, however, for a message's sender to be cognizant of whether a message's intended recipient implements privacy enhancements, in order that encoding and possible encipherment will not be performed on a message whose destination is not equipped to perform corresponding inverse transformations.

3. The defined mechanisms offer compatibility with a range of mail transport facilities (MTAs). Within the Internet, electronic mail transport is effected by a variety of SMTP implementations. Certain sites, accessible via SMTP, forward mail into other mail processing environments (e.g., USENET, CSNET, BITNET). The privacy enhancements must be able to operate across the SMTP realm; it is desirable that they also be compatible with protection of electronic mail sent between the SMTP environment and other connected environments.

4. The defined mechanisms offer compatibility with a broad range of electronic mail user agents (UAs). A large variety of electronic mail user agent programs, with a corresponding broad range of user interface paradigms, is used in the Internet. In order that an electronic mail privacy enhancement be available to the broadest possible user community, it is desirable that the selected mechanism be usable with the widest possible variety of existing UA programs. For purposes of pilot implementation, it is desirable that privacy enhancement processing be incorporable into a separate program, applicable to a range of UAs, rather than requiring internal modifications to

each UA with which enhanced privacy services are to be provided.

5. The defined mechanisms allow electronic mail privacy enhancement processing to be performed on personal computers (PCs) separate from the systems on which UA functions are implemented. Given the expanding use of PCs and the limited degree of trust which can be placed in UA implementations on many multi-user systems, this attribute can allow many users to process privacy-enhanced mail with a higher assurance level than a strictly UA-based approach would allow.

6. The defined mechanisms support privacy protection of electronic mail addressed to mailing lists.

In order to achieve applicability to the broadest possible range of Internet hosts and mail systems, and to facilitate pilot implementation and testing without the need for prior modifications throughout the Internet, three basic restrictions are imposed on the set of measures to be considered in this RFC:

1. Measures will be restricted to implementation at endpoints and will be amenable to integration at the user agent (UA) level or above, rather than necessitating integration into the message transport system (e.g., SMTP servers).

2. The set of supported measures enhances rather than restricts user capabilities. Trusted implementations, incorporating integrity features protecting software from subversion by local users, cannot be assumed in general. In the absence of such features, it appears more feasible to provide facilities which enhance user services (e.g., by protecting and authenticating inter-user traffic) than to enforce restrictions (e.g., inter-user access control) on user actions.

3. The set of supported measures focuses on a set of functional capabilities selected to provide significant and tangible benefits to a broad user community. By concentrating on the most critical set of services, we aim to maximize the added privacy value that can be provided with a modest level of implementation effort.

As a result of these restrictions, the following facilities can be provided:

-- disclosure protection,

-- sender authenticity, and

-- message integrity measures,

but the following privacy-relevant concerns are not addressed:

-- access control,

-- traffic flow security,

-- address list accuracy,

-- routing control,

-- issues relating to the serial reuse of PCs by multiple users,

-- assurance of message receipt and non-deniability of receipt, and

-- automatic association of acknowledgments with the messages to which they refer

An important goal is that privacy enhancement mechanisms impose a minimum of burden on the users they serve. In particular, this goal suggests eventual automation of the key management mechanisms supporting message encryption and authentication. In order to facilitate deployment and testing of pilot privacy enhancement implementations in the near term, however, compatibility with out-of-band (e.g., manual) key distribution must also be supported.

A message's sender will determine whether privacy enhancements are to be performed on a particular message. This will necessitate mechanisms by which a sender can determine whether particular recipients are equipped to process privacy-enhanced mail. In a general architecture, these mechanisms will be based on server queries; thus, the query function could be integrated into a UA to avoid imposing burdens or inconvenience on electronic mail users.

4 Processing of Messages

4.1 Message Processing Overview

This subsection provides a high-level overview of the components and processing steps involved in electronic mail privacy enhancement processing. Subsequent subsections will define the procedures in more detail.

A two-level keying hierarchy is used to support privacy-enhanced message transmission:

1. Data Encrypting Keys (DEKs) are used for encryption of message

text and for computation of message authentication codes (MACs). DEKs are generated individually for each transmitted message; no predistribution of DEKs is needed to support privacy-enhanced message transmission.

2. Interchange Keys (IKs) are used to encrypt DEKs for transmission. An IK may either be a single symmetric cryptographic key or, where asymmetric (public-key) cryptography is used for DEK encryption, the composition of a public component used by an originator and a secret component used by a recipient. Ordinarily, the same IK will be used for all messages sent between a given originator-recipient pair over a period of time. Each transmitted message includes a representation of the DEK(s) used for message encryption and/or authentication, encrypted under an individual IK per named recipient. This representation is accompanied by an identifier (IK ID) to enable the recipient to determine which IK was used, and so to decrypt the representation yielding the DEK required for message text decryption and/or MAC verification.

An encoding procedure is employed in order to represent encrypted message text in a universally transmissible form and to enable messages encrypted on one type of system to be decrypted on a different type. Four phases are involved in this process. A plaintext message is accepted in local form, using the host's native character set and line representation. The local form is converted to a canonical message text representation, defined as equivalent to the inter-SMTP representation of message text. The canonical representation is padded to an integral multiple of eight octets, as required by the encryption algorithm. MAC computation is performed, and (if disclosure protection is required), the padded canonical representation is encrypted. The output of this step is encoded into a printable form. The printable form is composed of a restricted character set which is chosen to be universally representable across sites, and which will not be disrupted by processing within and between MTS entities.

The output of the encoding procedure is combined with a set of header fields (to be defined in Section 4.8) carrying cryptographic control information. The result is passed to the electronic mail system to be encapsulated as the text portion of a transmitted message.

When a privacy-enhanced message is received, the cryptographic control fields within its text portion provide the information required for the authorized recipient to perform MAC verification and decryption on the received message text. First, the printable encoding is converted to a bitstring. If the transmitted message was encrypted, it is decrypted into the canonical representation. If the message was not encrypted, decoding from the printable form produces the canonical representation directly. The MAC is verified, and the

canonical representation is converted to the recipient's local form, which need not be the same as the sender's local form.

4.2 Encryption Algorithms and Modes

For purposes of this RFC, the Block Cipher Algorithm DEA-1, defined in ISO draft international standard DIS 8227 [1] shall be used for encryption of message text and for computation of authentication codes on messages. The DEA-1 is equivalent to the Data Encryption Standard (DES), as defined in FIPS PUB 46 [2]. When used for these purposes, the DEA-1 shall be used in the Cipher Block Chaining (CBC) mode, as defined in ISO DIS 8372 [3]. The CBC mode definition in DIS 8372 is equivalent to that provided in FIPS PUB 81 [4]. A unique initializing vector (IV) will be generated for and transmitted with each encrypted electronic mail message.

An algorithm other than DEA-1 may be employed, provided that it satisfies the following requirements:

1. it must be a 64-bit block cipher, enciphering and deciphering in 8 octet blocks

2. it is usable in the ECB and CBC modes defined in DIS8372

3. it is able to be keyed using the procedures and parameters defined in this RFC

4. it is appropriate for MAC computation

5. cryptographic key field lengths are limited to 16 octets in length

Certain operations require that one key be encrypted under another key (interchange key) for purposes of transmission. For purposes of this RFC, such encryption will be performed using DEA-1 in Electronic Codebook (ECB) mode. An optional facility is available to an interchange key provider to indicate that an associated key is to be used for encryption in another mode (e.g., the Encrypt-Decrypt-Encrypt (EDE) mode used for key encryption and decryption with pairs of 64-bit keys, as described [5] by ASC X3T1).

Future support of public key algorithms for key encryption is under consideration, and it is intended that the procedures defined in this RFC be appropriate to allow such usage. Support of key encryption modes other than ECB is optional for implementations, however. Therefore, in support of universal interoperability, interchange key providers should not specify other modes in the absence of a priori information indicating that recipients are equipped to perform key encryption in other modes.

4.3 Canonical Encoding

Any encryption scheme must be compatible with the transparency constraints of its underlying electronic mail facilities. These constraints are generally established based on expected user requirements and on the characteristics of anticipated endpoint transport facilities. SMTP, designed primarily for interpersonal messages and anticipating systems and transport media which may be restricted to a 7-bit character set, can transmit any 7-bit characters (but not arbitrary 8-bit binary data) in message text.

SMTP introduces other transparency constraints related to line lengths and message delimiters. Message text may not contain the string "<CR><LF>.<CR><LF>" in sequence before the end of a message, and must contain the string "<CR><LF>" at least every 1000 characters. Another important SMTP transparency issue must be noted. Although SMTP specifies a standard representation for line delimiters (ASCII <CR><LF>), numerous systems use a different native representation to delimit lines. For example, the <CR><LF> sequences delimiting lines in mail inbound to UNIX(tm) systems are transformed to single <LF>s as mail is written into local mailbox files. Lines in mail incoming to record-oriented systems (such as VAX VMS) may be converted to appropriate records by the destination SMTP [6] server. As a result, if the encryption process generated <CR>s or <LF>s, those characters might not be accessible to a recipient UA program at a destination using different line delimiting conventions. It is also possible that conversion between tabs and spaces may be performed in the course of mapping between inter-SMTP and local format; this is a matter of local option. If such transformations changed the form of transmitted ciphertext, decryption would fail to regenerate the transmitted plaintext, and a transmitted MAC would fail to compare with that computed at the destination.

The conversion performed by an SMTP server at a system with EBCDIC as a native character set has even more severe impact, since the conversion from EBCDIC into ASCII is an information-losing transformation. In principle, the transformation function mapping between inter-SMTP canonical ASCII message representation and local format could be moved from the SMTP server up to the UA, given a means to direct that the SMTP server should no longer perform that transformation. This approach has the disadvantage that it would imply internal file (e.g., mailbox) formats which would be incompatible with the systems on which they reside, an untenable prospect. Further, it would require modification to SMTP servers, as mail would be passed to SMTP in a different representation than it is passed at present.

Our approach to this problem selects a canonical character set, uniformly representable across privacy-enhanced UAs regardless of their systems' native character sets, to transport encrypted mail text (but not electronic mail transport headers!) between endpoints.

In this approach, an outbound privacy-enhanced message is transformed between four forms, in sequence:

1. (Local_Form) The message text is created (e.g., via an editor) in the system's native character set, with lines delimited in accordance with local convention.

2. (Canonicalize) The message text is converted to the universal canonical form, equivalent to the inter-SMTP representation as defined in RFC822 [7] (ASCII character set, <CR><LF> line delimiters). (The processing required to perform this conversion is relatively small, at least on systems whose native character set is ASCII.)

3. (Encipher/Authenticate) A padded version of the canonical plaintext representation is created by appending zero-valued octets to the end of the representation until the length is an integral multiple of 8 octets, as is required to perform encryption in the DEA-1 CBC mode. No padding is applied if the canonical plaintext representation's length is already a multiple of 8 octets. This padded representation is used as the input to the encryption function and to the MAC computation function.

4. (Encode to Printable Form) The bits resulting from the encryption operation are encoded into characters which are universally representable at all sites, though not necessarily with the same bit patterns (e.g., although the character "E" is represented in an ASCII-based system as hexadecimal 45 and as hexadecimal C5 in an EBCDIC-based system, the local significance of the two representations is equivalent). Use of a 64-character subset of International Alphabet IA5 is proposed, enabling 6 bits to be represented per printable character. (The proposed subset of characters is represented identically in IA5 and ASCII.) Two additional characters, "=" and "*", are used to signify special processing functions. The encoding function's output is delimited into text lines (using local conventions), with each line containing 64 printable characters. The encoding process is performed as follows, transforming strings of 3 arbitrary (8-bit) characters to strings of 4 encoded characters:

   4a. Proceeding from left to right across the input characters (considered as a contiguous bitstring), each group of 6 bits is used as an index into an array of 64 printable characters; the character referenced by the index is placed in the output string. These characters, identified in Table 1, are selected so as to be universally representable, and the set excludes characters with particular significance to SMTP e.g.,

".", "<CR>", "<LF>").

4b. If fewer than 3 input characters are available in a final quantum, zero bits are added (on the right) to form an integral number of 6-bit groups. Output character positions which are not required to represent actual input data are set to a 65th reserved, universally representable character ("="). Use of a reserved character for padding allows compensatory processing to be performed by a recipient, allowing the decoded message text's length to be precisely the same as the input message's length. A final 3-octet input quantum will be represented as a 4 octet encoding with no terminal "=", a 2-octet input quantum will be represented as 3 octets followed by one terminal "=", and a 1-octet input quantum will be represented as 2 octets followed by two occurrences of "=".

A sender may exclude one or more portions of a message from encryption/authentication processing. Explicit action is required to exclude a portion of a message from such processing; by default, encryption/authentication is applied to the entirety of message text. The user-level delimiter which specifies such exclusion is a local matter, and hence may vary between sender and recipient, but all systems should provide a means for unambiguous identification of areas excluded from encryption/authentication processing. An excluded area is represented in the inter-SMTP transmission form (universal across communicating systems) by bracketing with the reserved delimiter "*". Cryptographic state is preserved transparently across an excluded area and continued after the end of the excluded area. A printable encoding quantum (per step 4b) is completed before the delimiter "*" is output to initiate or terminate the representation of an excluded block. Note that the canonicalizing transformation (step 2 above) and the encoding to printable form (step 4 above) are applied to all portions of message text, even those excluded from encryption and authentication.

In summary, the outbound message is subjected to the following composition of transformations:

```
Transmit_Form = Encode(Encipher(Canonicalize(Local_Form)))
```

The inverse transformations are performed, in reverse order, to process inbound privacy-enhanced mail:

```
Local_Form = DeCanonicalize(Decipher(Decode(Transmit_Form)))
```

Note that the local form and the functions to transform messages to and from canonical form may vary between the sender and recipient systems without loss of information.

Value	Encoding	Value	Encoding	Value	Encoding	Value	Encoding
0	A	17	R	34	i	51	z
1	B	18	S	35	j	52	0
2	C	19	T	36	k	53	1
3	D	20	U	37	l	54	2
4	E	21	V	38	m	55	3
5	F	22	W	39	n	56	4
6	G	23	X	40	o	57	5
7	H	24	Y	41	p	58	6
8	I	25	Z	42	q	59	7
9	J	26	a	43	r	60	8
10	K	27	b	44	s	61	9
11	L	28	c	45	t	62	+
12	M	29	d	46	u	63	/
13	N	30	e	47	v		
14	O	31	f	48	w	(pad)	=
15	P	32	g	49	x		
16	Q	33	h	50	y	(1)	*

(1) The character "*" is used to delimit portions of an encoded message to which encryption/authentication processing has not been applied.

Printable Encoding Characters
Table 1

4.4 Encapsulation Mechanism

Encapsulation of privacy-enhanced messages within an enclosing layer of headers interpreted by the electronic mail transport system offers a number of advantages in comparison to a flat approach in which certain fields within a single header are encrypted and/or carry cryptographic control information. Encapsulation provides generality and segregates fields with user-to-user significance from those transformed in transit. As far as the MTS is concerned, information incorporated into cryptographic authentication or encryption processing will reside in a message's text portion, not its header portion.

The encapsulation mechanism to be used for privacy-enhanced mail is derived from that described in RFC934 [8] which is, in turn, based on precedents in the processing of message digests in the Internet community. To prepare a user message for encrypted or authenticated transmission, it will be transformed into the representation shown in Figure 1. Note that, while encryption and/or authentication processing of transmitted mail may depend on information contained in the enclosing header (e.g., "To:"), all fields inserted in the course of encryption/authentication processing are placed in the encapsulated header. This facilitates compatibility with mail handling programs which accept only text, not header fields, from input files or from other programs. Further, privacy enhancement

processing can be applied recursively.

Sensitive data should be protected by incorporating the data within the encapsulated text rather than by applying measures selectively to fields in the enclosing header. Examples of potentially sensitive header information may include fields such as "Subject:", with contents which are significant on an end-to-end, inter-user basis. The (possibly empty) set of headers to which protection is to be applied is a user option. If an authenticated version of header information is desired, that data can be replicated within the encapsulated text portion in addition to its inclusion in the enclosing header. If a user wishes disclosure protection for header fields, they must occur only in the encapsulated text and not in the enclosing or encapsulated header. If disclosure protection is desired for the "Subject:" field, it is recommended that the enclosing header contain a "Subject:" field indicating that "Encrypted Mail Follows".

A specific point regarding the integration of privacy-enhanced mail facilities with the message encapsulation mechanism is worthy of note. The subset of IA5 selected for transmission encoding intentionally excludes the character "-", so encapsulated text can be distinguished unambiguously from a message's closing encapsulation boundary (Post-EB) without recourse to character stuffing.

4.5 Processing for Authentication Without Confidentiality

When a message is to be authenticated without confidentiality service, a DEK is generated [9] for use in MAC computation, and a MAC is computed using that DEK. For each individually identified recipient, an IK is selected and identified with an "X-IK-ID:" field. Each "X-IK-ID:" field is followed by an "X-Key-Info:" field which transfers the key under which MAC computation was performed, encrypted under the IK identified by the preceding "X-IK-ID:" field, along with a representation of the MAC encrypted under the same IK. The encapsulated text portion following the encapsulated header is canonically encoded, and coded into printable characters for transmission, but not encrypted.

Enclosing Header Portion

(Contains header fields per RFC-822)

Blank Line

(Separates Enclosing Header from Encapsulated Message)

Encapsulated Message

Pre-Encapsulation Boundary (Pre-EB)

-----PRIVACY-ENHANCED MESSAGE BOUNDARY-----

Encapsulated Header Portion

(Contains encryption control fields inserted in plaintext. Examples include "X-IV:", "X-IK-ID:", "X-Key-Info:", and "X-Pad-Count:". Note that, although these control fields have line-oriented representations similar to RFC-822 header fields, the set of fields valid in this context is disjoint from those used in RFC-822 processing.)

Blank Line

(Separates Encapsulated Header from subsequent encoded Encapsulated Text Portion)

Encapsulated Text Portion

(Contains message data encoded as specified in Section 4.3; may incorporate protected copies of "Subject:", etc.)

Post-Encapsulation Boundary (Post-EB)

-----PRIVACY-ENHANCED MESSAGE BOUNDARY-----

Message Encapsulation
Figure 1

4.6 Processing for Authentication and Confidentiality

When a message is to be authenticated with confidentiality service, a DEK is generated for use in MAC computation and a variant of the DEK is formed for use in message encryption. For each individually identified recipient, an IK is selected and identified with an "X-IK-ID:" field. Each "X-IK-ID:" field is followed by an "X-Key-Info:" field, which transfers the DEK and computed MAC, each encrypted under the IK identified in the preceding "X-IK-ID:" field. The encapsulated text portion following the encapsulated header is canonically encoded, encrypted, and coded into printable characters

for transmission.

4.7 Mail for Mailing Lists

When mail is addressed to mailing lists, two different methods of processing can be applicable: the IK-per-list method and the IK-per-recipient method. The choice depends on the information available to the sender and on the sender's preference.

If a message's sender addresses a message to a list name or alias, use of an IK associated with that name or alias as a entity (IK-per-list), rather than resolution of the name or alias to its constituent destinations, is implied. Such an IK must, therefore, be available to all list members. This alternative will be the normal case for messages sent via remote exploder sites, as a sender to such lists may not be cognizant of the set of individual recipients. Unfortunately, it implies an undesirable level of exposure for the shared IK, and makes its revocation difficult. Moreover, use of the IK-per-list method allows any holder of the list's IK to masquerade as another sender to the list for authentication purposes.

If, in contrast, a message's sender is equipped to expand the destination mailing list into its individual constituents and elects to do so (IK-per-recipient), the message's DEK and MAC will be encrypted under each per-recipient IK and all such encrypted representations will be incorporated into the transmitted message. (Note that per-recipient encryption is required only for the relatively small DEK and MAC quantities carried in the X-Key-Info field, not for the message text which is, in general, much larger.) Although more IKs are involved in processing under the IK-per-recipient method, the pairwise IKs can be individually revoked and possession of one IK does not enable a successful masquerade of another user on the list.

4.8 Summary of Added Header and Control Fields

This section summarizes the syntax and semantics of the new header and control fields to be added to messages in the course of privacy enhancement processing, indicating whether a particular field occurs in a message's encapsulated header portion or its encapsulated text portion. Figure 2 shows the appearance of a small example encapsulated message using these fields. In all cases, hexadecimal quantities are represented as contiguous strings of digits, where each digit is represented by a character from the ranges "0"-"9" or upper case "A"-"F". Unless otherwise specified, all arguments are to be processed in a case-sensitive fashion.

Although the encapsulated header fields resemble RFC-822 header fields, they are a disjoint set and will not in general be processed by the same parser which operates on enclosing header fields. The complexity of lexical analysis needed and appropriate for

encapsulated header field processing is significantly less than that appropriate to RFC-822 header processing. For example, many characters with special significance to RFC-822 at the syntactic level have no such significance within encapsulated header fields.

The "X-IK-ID" and "X-Key-Info" fields are the only encapsulated header fields with lengths which can vary beyond a size conveniently printable on a line. Whitespace may be used between the subfields of these fields to fold them in the manner of RFC-822; such whitespace is not to be interpreted as a part of a subfield.

```
-----PRIVACY-ENHANCED MESSAGE BOUNDARY-----
X-Proc-Type: 1,E
X-Pad-Count: 1
X-IV: F8143EDE5960C597
X-IK-ID: JL:3:ECB
X-Key-Info: 9FD3AAD2F2691B9A,B70665BB9BF7CBCD
X-IK-ID: JL:1:ECB
X-Key-Info: 161A3F75DC82EF26,E2EF532C65CBCFF7

LLrHB0eJzyhP+/fSStdW8okeEnv47jxe7SJ/iN72ohNcUk2jHEUSoHlnvNSIWL9M
8tEjmF/zxB+bATMtPjCUWbz8Lr9wloXIkjHUlBLpvXR0UrUzYbkNpk0agV2IzUpk
J6UiRRGcDSvzrsoK+oNvqu6z7Xs5Xfz5rDqUcMlKlZ6720dcBWGGsDLpTpSCnpot
dXd/H5LMDWnonNvPCwQUHt==
-----PRIVACY-ENHANCED MESSAGE BOUNDARY-----
```

Example Encapsulated Message

Figure 2

X-IK-ID: This field is placed in the encapsulated header portion of a message to identify the Interchange Key used for encryption of an associated Data Encrypting Key or keys (used for message text encryption and/or MAC computation). This field is used for messages authenticated without confidentiality service and for messages authenticated with confidentiality service. The field contains (in order) an Issuing Authority subfield and an IK Qualifier subfield, and may also contain an optional IK Use Indicator subfield. The subfields are delimited by the colon character (":"), optionally followed by whitespace. Section 5.1.2, Interchange Keys, discusses the semantics of these subfields and specifies the alphabet from which they are chosen. Note that multiple X-IK-ID fields may occur within a single encapsulated header. Each X-IK-ID field is associated with an immediately subsequent X-Key-Info field.

X-IV: This field is placed in the encapsulated header portion of a message to carry the Initializing Vector

used for message encryption. It is used only for messages where confidentiality service is applied. Following the field name, and one or more delimiting whitespace characters, a 64-bit Initializing Vector is represented as a contiguous string of 16 hexadecimal digits.

X-Key-Info: This field is placed in a message's encapsulated header portion to transfer two items: a DEK and a MAC. Both items are encrypted under the IK identified by a preceding X-IK-ID field; they are represented as two strings of contiguous hexadecimal digits, separated by a comma. For DEA-1, the DEK representation will be 16 hexadecimal digits (corresponding to a 64-bit key); this subfield can be extended to 32 hexadecimal digits (corresponding to a 128-bit key) if required to support other algorithms. The MAC is a 64-bit quantity, represented as 16 hexadecimal digits. The MAC is computed under an unmodified version of the DEK. Message encryption is performed using a variant of the DEK, formed by modulo-2 addition of the hexadecimal quantity F0F0F0F0F0F0F0F0 to the DEK.

X-Pad-Count: This field is placed in the encapsulated header portion of a message to indicate the number of zero-valued octets which were added to pad the input stream to the encryption function to an integral multiple of eight octets, as required by the DEA-1 CBC encryption mode. A decimal number in the range 0-7 follows the field name, and one or more delimiting whitespace characters. Inclusion of this field allows disambiguation between terminal zero-valued octets in message text (admittedly, a relatively unlikely prospect) and zero-valued octets inserted for padding purposes.

X-Proc-Type: This field is placed in the encapsulated header portion of a message to identify the type of processing performed on the transmitted message. The first subfield is a decimal version number, which will be used if future developments make it necessary to redefine the interpretation of encapsulated header fields. At present, this field may assume only the value "1". The second subfield, delimited by a comma, assumes one of two single-character alphabetic values: "A" and "E", to signify, respectively, (1) authentication processing only and (2) the combination of authentication and confidentiality service through encryption.

5 Key Management

5.1 Types of Keys

5.1.1 Data Encrypting Keys (DEKs)

Data Encrypting Keys (DEKs) are used for encryption of message text and for computation of message authentication codes (MACs). It is strongly recommended that DEKs be generated and used on a one-time basis. A transmitted message will incorporate a representation of the DEK encrypted under an interchange key (IK) known to the authorized recipient.

DEK generation can be performed either centrally by key distribution centers (KDCs) or by endpoint systems. One advantage of centralized KDC-based generation is that DEKs can be returned to endpoints already encrypted under the IKs of message recipients. This reduces IK exposure and simplifies endpoint key management requirements. Further, dedicated KDC systems may be able to implement better algorithms for random key generation than can be supported in endpoint systems. On the other hand, decentralization allows endpoints to be relatively self-sufficient, reducing the level of trust which must be placed in components other than a message's originator and recipient. Moreover, decentralized DEK generation by endpoints reduces the frequency with which senders must make real-time queries of (potentially unique) servers in order to send mail, enhancing communications availability.

5.1.2 Interchange Keys (IKs)

Interchange Keys (IKs) are used to encrypt Data Encrypting Keys. In general, the granularity of IK usage is at the pairwise per-user level except for mail sent to address lists comprising multiple users. In order for two principals to engage in a useful exchange of privacy-enhanced electronic mail using conventional cryptography, they must first share a common interchange key. When asymmetric cryptography is used, an originator and recipient must possess appropriate public and secret components which, in composition, constitute an interchange key.

The means by which interchange keys are provided to appropriate parties are outside the scope of this RFC, but may be centralized (e.g., via key management servers) or decentralized (e.g., via direct distribution among users). In any case, a given IK is associated with a responsible Issuing Authority (IA). When an IA generates and distributes an IK, associated control information must be provided to direct how that IK is to be used. In order to select the appropriate IK to use in message encryption, a sender must retain a correspondence between IKs and the recipients with which they are associated. Expiration date information must also be retained, in order that cached entries may be invalidated and replaced as

appropriate.

When a privacy-enhanced message is transmitted, an indication of the IK (or IKs, in the case of a message sent to multiple recipients) used for DEK encryption must be included. To this end, the IK ID construct is defined to provide the following data:

1. Identification of the relevant Issuing Authority (IA subfield)

2. Qualifier string to distinguish the particular IK within the set of IKs distributed by the IA (IK qualifier subfield)

3. (Optional) Indicator of IK usage mode (IK use indicator subfield)

The subfields of an IK ID are delimited with the colon character (":"). The IA and IK qualifier subfields are generated from a restricted character set, as prescribed by the following BNF (using notation as defined in RFC-822, sections 2 and 3.3):

```
IAorIKQual    :=    1*ia-char

ia-char       :=    DIGIT / ALPHA / "'" / "+" / "(" / ")" /
                    "," / "." / "/" / "=" / "?" / "-" / "@" /
                    "%" / "!" / '"' / "_" / "<" / ">"
```

The IK use indicator subfield assumes a value from a small set of reserved strings, described later in this section.

IA identifiers must be assigned in a manner which assures uniqueness. This can be done on a centralized or hierarchic basis.

The IK qualifier string format may vary among different IAs, but must satisfy certain functional constraints. An IA's IK qualifiers must be sufficient to distinguish among the set of IKs issued by that IA. Since a message may be sent with multiple IK IDs, corresponding to multiple intended recipients, each recipient must be able to determine which IK is intended for it. Moreover, if no corresponding IK is available in the recipient's database when a message arrives, the recipient must be able to determine which IK to request and to identify that IK's associated IA. Note that different IKs may be used for different messages between a pair of communicants. Consider, for example, one message sent from A to B and another message sent (using the IK-per-list method) from A to a mailing list of which B is a member. The first message would use an IK shared between A and B, but the second would use an IK shared among list members.

While use of a monotonically increasing number as an IK qualifier is sufficient to distinguish among the set of IKs distributed by an IA, it offers no facility for a recipient lacking a matching IK to determine the appropriate IK to request. This suggests that sender and recipient name information should be incorporated into an IK qualifier, along with a number to distinguish among multiple IKs used between a sender/recipient pair. In order to support universal interoperability, it is necessary to assume a universal form for the naming information. General definition of such a form requires further study; issues and possible approaches will be noted in Section 6. As an interim measure, the following IK qualifier format is suggested:

<sender-name>/<recipient-name>/<numid>

where <sender-name> and <recipient-name> are in the following form:

<user>@<domain-qualified-host>

For the case of installations which transform local host names before transmission into the broader Internet, it is strongly recommended that the host name as presented to the Internet be employed. The <numid> is a contiguous string of decimal digits.

The IK use indicator subfield is an optional facility, provided to identify the encryption mode in which the IK is to be used. Currently, this subfield may assume the following reserved string values: "ECB" and "EDE"; the default value is ECB.

An example IK ID adhering to this recommendation is as follows:

ptf-kmc:linn@CCY.BBN.COM/privacy-tf@C.ISI.EDU/2:ECB

This IK ID would indicate that IA "ptf-kmc" has issued an IK for use on messages sent from "linn@CCY.BBN.COM" to "privacy-tf@C.ISI.EDU", that the IA has associated number 2 with that IK, and that the IK is to be used in ECB mode.

IKs will remain valid for a period which will be longer than a single message and will be identified by an expiration time distributed along with the IK; IK cryptoperiod is dictated in part by a tradeoff between key management overhead and revocation responsiveness. It would be undesirable to delete an IK permanently before receipt of a message encrypted using that IK, as this would render the message permanently undecipherable. Access to an expired IK would be needed, for example, to process mail received by a user (or system) which had been inactive for an extended period of time. In order to enable very old IKs to be deleted, a message's recipient desiring encrypted local long term storage should transform the DEK used for message text encryption via re-encryption under a locally maintained IK, rather than relying on IA maintenance of old IKs for indefinite

periods.

6 User Naming

Unique naming of electronic mail users, as is needed in order to select corresponding keys correctly, is an important topic and one requiring significant study. A logical association exists between key distribution and name/directory server functions; their relationship is a topic deserving further consideration. These issues have not been fully resolved at this writing. The interim architecture relies on association of IKs with user names represented in a universal form, which has the following properties:

1. The universal form must be specifiable by an IA as it distributes IKs and known to a UA as it processes received IKs and IK IDs. If a UA or IA uses addresses in a local form which is different from the universal form, it must be able to perform an unambiguous mapping from the universal form into the local representation.

2. The universal form, when processed by a sender UA, must have a recognizable correspondence with the form of a recipient address as specified by a user (perhaps following local transformation from an alias into a universal form)

It is difficult to ensure these properties throughout the Internet. For example, an MTS which transforms address representations between the local form used within an organization and the global form used for Internet mail transmission may cause property 2 to be violated.

The use of flat (non-hierarchic) electronic mail user identifiers, which are unrelated to the hosts on which the users reside, appears useful. Personal characteristics, like social security numbers, might be considered. Individually-selected identifiers could be registered with a central authority, but a means to resolve name conflicts would be necessary.

A point of particular note is the desire to accommodate multiple names for a single individual, in order to represent and allow delegation of various roles in which that individual may act. A naming mechanism that binds user roles to keys is needed. Bindings cannot be immutable since roles sometimes change (e.g., the comptroller of a corporation is fired).

It may be appropriate to examine the prospect of extending the Domain Name System and its associated name servers to resolve user names to unique user IDs. An additional issue arises with regard to mailing list support: name servers do not currently perform (potentially recursive) expansion of lists into users. ISO and CSNet are working on user-level directory service mechanisms, which may also bear

consideration.

7 Example User Interface and Implementation

In order to place the mechanisms and approaches discussed in this RFC into context, this section presents an overview of a prototype implementation. This implementation is a standalone program [10] which is invoked by a user, and lies above the existing UA sublayer. This form of integration offers the advantage that the program can be used in conjunction with a range of UA programs, rather than being compatible only with a particular UA. When a user wishes to apply privacy enhancements to an outgoing message, the user prepares the message's text and invokes the standalone program (interacting with the program in order to provide address information and other data required to perform privacy enhancement processing), which in turn generates output suitable for transmission via the UA. When a user receives a privacy-enhanced message, the UA delivers the message in encrypted form, suitable for decryption and associated processing by the standalone program.

In this prototype implementation, a cache of IKs is maintained in a local file, with entries managed manually based on pairwise agreements between originators and recipients. This cache is, effectively, a simple database. IKs are selected for transmitted messages based on recipient names, and corresponding IK IDs are placed into the message's encapsulated header. When a message is received, the IK ID is used as a basis for a lookup in the database, yielding the appropriate IK entry. DEKs and IVs are generated dynamically within the program.

Options (e.g., authentication only vs. authentication with confidentiality service) are selected by command line arguments to the standalone program. Destination addresses are specified in the same fashion. The function of specifying destination addresses to the privacy enhancement program is logically distinct from the function of specifying the corresponding addresses to the UA for use by the MTS. This separation results from the fact that, in many cases, the local form of an address as specified to a UA differs from the Internet global form as used for IK ID fields.

8 Areas For Further Study

The procedures defined in this RFC are sufficient to support pilot implementation of privacy-enhanced electronic mail transmission among cooperating parties in the Internet. Further effort will be needed, however, to enhance robustness, generality, and interoperability. In particular, further work is needed in the following areas:

1. User naming techniques, and their relationship to the domain system, name servers, directory services, and key management

functions

2. Standardization of Issuing Authority functions, including protocols for communications among IAs and between User Agents and IAs

3. Use of public key encryption algorithms to encrypt data encrypting keys

4. Interoperability with X.400 mail

We anticipate generation of subsequent RFCs which will address these topics.

9 References

This section identifies background references which may be useful to those contemplating use of the mechanisms defined in this RFC.

ISO 7498/Part 2 - Security Architecture, prepared by ISO.TC97/SC 21/WG 1 Ad hoc group on Security, extends the OSI Basic Reference Model to cover security aspects which are general architectural elements of communications protocols, and provides an annex with tutorial and background information.

US Federal Information Processing Standards Publication (FIPS PUB) 46, Data Encryption Standard, 15 January 1977, defines the encipherment algorithm used for message text encryption and MAC computation.

FIPS PUB 81, DES Modes of Operation, 2 December 1980, defines specific modes in which the Data Encryption Standard algorithm is to be used to perform encryption and MAC computation.

NOTES:

[1] Information Processing Systems: Data Encipherment: Block Cipher Algorithm DEA 1.

[2] Federal Information Processing Standards Publication 46, Data Encryption Standard, 15 January 1977.

[3] Information Processing Systems: Data Encipherment: Modes of Operation of a 64-bit Block Cipher

[4] Federal Information Processing Standards Publication 81, DES Modes of Operation, 2 December 1980.

[5] Addendum to the Transport Layer Protocol Definition for Providing Connection Oriented End to End Cryptographic Data Protection Using a 64-Bit Block Cipher, X3T1-85-50.3, draft of 19 December 1985, Gaithersburg, MD, p. 15.

[6] This transformation should occur only at an SMTP endpoint, not at an intervening relay, but may take place at a gateway system linking the SMTP realm with other environments.

[7] Crocker, D. Standard for the Format of ARPA Internet Text Messages (RFC822), August 1982.

[8] Rose, M. T., and Stefferud, E. A., Proposed Standard for Message Encapsulation, January 1985.

[9] Key generation for authentication and message text encryption may either be performed by the sending host or by a centralized server. This RFC does not constrain this design alternative. Section 5.1.1 identifies possible advantages of a centralized server approach.

[10] Note that in the UNIX(tm) system, and possibly in other environments as well, such a program can be invoked as a "filter" within an electronic mail UA or a text editor, simplifying the sequence of operations which must be performed by the user.